

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 1 月 2 4 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 0 1 6 6 3 8
Application Number:

[ST. 10/C]: [J P 2 0 0 3 - 0 1 6 6 3 8]

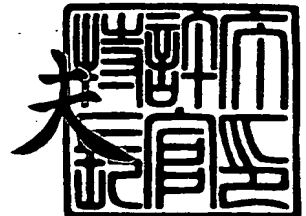
出 願 人 株式会社セガ
Applicant(s): 任天堂株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2 0 0 4 年 2 月 2 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 P020064

【提出日】 平成15年 1月24日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/320

【発明者】

【住所又は居所】 東京都大田区東糀谷 2 - 1 2 - 1 4 株式会社アミューズメントヴィジョン内

【氏名】 桂川 昇治

【発明者】

【住所又は居所】 東京都大田区東糀谷 2 - 1 2 - 1 4 株式会社アミューズメントヴィジョン内

【氏名】 加来 徹也

【発明者】

【住所又は居所】 東京都大田区東糀谷 2 - 1 2 - 1 4 株式会社アミューズメントヴィジョン内

【氏名】 大山 和紀

【発明者】

【住所又は居所】 東京都大田区東糀谷 2 - 1 2 - 1 4 株式会社アミューズメントヴィジョン内

【氏名】 鈴木 良則

【特許出願人】

【識別番号】 000132471

【氏名又は名称】 株式会社セガ

【特許出願人】

【識別番号】 000233778

【氏名又は名称】 任天堂株式会社

【代理人】

【識別番号】 100094514

【弁理士】

【氏名又は名称】 林 恒徳

【選任した代理人】

【識別番号】 100094525

【弁理士】

【氏名又は名称】 土井 健二

【手数料の表示】

【予納台帳番号】 030708

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0018629

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ゲームのキャラクタに画像を投影するプログラム、そのプログラムが組み込まれたゲーム機、及びそのプログラムが格納された記録媒体

【特許請求の範囲】

【請求項 1】 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有するゲーム機において、ゲームのキャラクタに所定の画像を投影するプログラムであって、

前記操作手段の操作により 2 次元座標からなる画像を前記制御手段により生成する画像作成ステップと、

前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付するステップと、
を実現させるためのプログラム。

【請求項 2】 前記キャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる、

ことを実現させるための請求項 1 記載のプログラム。

【請求項 3】 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有し、ゲームのキャラクタに所定の画像を投影するプログラムが組み込まれたゲーム機であって、

前記操作手段の操作により 2 次元座標からなる画像を前記制御手段により生成する画像作成ステップと、

前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付するステップと、
を実現させるためのプログラム。

ラクタに前記画像を投影した投影像を前記キャラクタに貼付するステップと、
を実現させるためのプログラムが組み込まれたゲーム機。

【請求項 4】 前記キャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる、

ことを実現させるための請求項 3 記載のプログラムが組み込まれたゲーム機。

【請求項 5】 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有するゲーム機に対してゲームのキャラクタに所定の画像を投影するプログラムが格納された記録媒体であって、

前記操作手段の操作により 2 次元座標からなる画像を前記制御手段により生成する画像作成ステップと、

前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクタ周囲の任意の位置に配置し、前記投影光源から前記キャラクタに前記画像を投影した投影像を前記キャラクタに貼付するステップと、

を実現させるためのプログラムが格納された記録媒体。

【請求項 6】 前記キャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる、

ことを実現させるための請求項 5 記載のプログラムが格納された記録媒体。

【請求項 7】 三次元仮想空間内において、三次元座標からなるオブジェクトに所定の投影画像を投影したオブジェクト画像をディスプレイ装置に表示するための画像データを生成する画像データ生成手段としてコンピュータを動作させる画像表示制御プログラムであって、

前記画像表示制御プログラムは、制御手段に、

前記オブジェクトに投影される所定の投影画像データを生成させ、

前記仮想空間ないに前記オブジェクトおよび前記投影画像を配置させ、
操作者の操作に基づいて前記オブジェクトに対する前記投影画像の相対位置と、
前記投影画像を前記オブジェクトに投影する仮想光源の位置を決定させ、
前記仮想光源と前記オブジェクト上の投影位置を含む投影面の距離を演算し、
前記仮想光源から所定距離以上はなれた投影面を投影対象から除いて、前記オブジェクトにおいて前記投影画像を投影すべき投影面を演算させ、
前記仮想光源と視点として前記投影画像を前記投影面に投影して、前記投影画像を前記オブジェクトの前記投影面に貼り付けさせ、
前記投影画像が貼り付けられた前記オブジェクトについてのオブジェクト画像データを生成させる機能を実現させることを特徴とする画像表示制御プログラム。

【請求項 8】 前記制御手段に、さらに前記投影画像の画像データをユーザーが操作手段から入力する操作信号に基づいて生成させる機能を実現させることを特徴とする請求項 7 記載の画像表示制御プログラム。

【請求項 9】 前記制御手段に、前記投影画像を前記オブジェクトと同一ないし近似する形状を備える透明オブジェクトに対して投影して、前記投影画像を前記透明オブジェクトに貼り付けさせ、

前記表示装置に前記投影画像を貼り付けた透明オブジェクトを前記オブジェクトに重ねて表示させる機能を実現させることを特徴とする請求項 7 又は 8 記載の画像表示制御プログラム。

【請求項 10】 前記仮想光源と前記投影オブジェクトと前記オブジェクトの位置関係によって、前記オブジェクトに貼り付けられた投影画像の変形が著しいと判断される場合、前記投影オブジェクトの透明度を変化させる機能を実現させることを特徴とする請求項 7 乃至 9 の画像表示制御プログラム。

【請求項 11】 前記オブジェクトが複数のパーツの組み合わせによって構成されており、

前記制御部に、操作者の操作に応じて前記パーツのうち少なくとも一つが前記投影画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせる機能を実現させることを特徴とする請求項 7 乃至 10 記載の

画像表示制御プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ゲームのキャラクタに所望の画面を自由に貼り付ける投影のプログラム、ゲームのキャラクタに画像を投影するプログラムが組み込まれたゲーム機、及びゲームのキャラクタに画像を投影するプログラムが格納された記録媒体に関する。

【0002】

【従来の技術】

従来から、ゲームのキャラクタに対してユーザーが所望の画面を貼り付けたり、キャラクタ自身の色や表示などを変えたりすることが行われている。

【0003】

例えば、ゲーム内でユーザーが使用するロボットなどのキャラクタにエンブレムと呼ばれる表示（例えば旗や動物などのイメージ）を所定の位置に貼り付けたりする技術がある。さらに、ユーザーがゲーム内で使用する車のキャラクタに色や模様を付したりする技術もある。ユーザー自身の使用するキャラクタに愛着を持たせるためである。

【0004】

これらは、例えばゲームの開始前に、メインメニューでの所定の選択によりエンブレムの選択画面やエンブレム作成画面、あるいは、色や模様を付したりするための作成画面が選択され、その画面内で作成、選択が行われる。そして、作成後のキャラクタをゲーム機内のメモリなどで保持しておき、ゲーム開始とともに呼び出すことで、ユーザー自身で作成或いは選択したエンブレム等付きのキャラクタが表示され、実際にゲームが開始されることになる。

【0005】

一方で、いわゆる3次元コンピュータグラフィックスで3次元モデルをレンダリングして2次元画像データを映り込ませることも行われている（例えば特許文献1）。

【0006】

【特許文献1】特許公開公報 特開2001-351121号

【0007】

【発明が解決しようとする課題】

しかし、従来では、エンブレムのキャラクタに対する貼り付け位置が固定されており、自身の作成等したエンブレムをキャラクタの様々な場所に自由に貼り付けることができなかった。

【0008】

キャラクタが固定された位置ではなく様々な位置にエンブレムを自由に貼り付けを行えるようにするために、例えば、固定された位置をそのまま移動させることも考えられる。しかし、移動した位置で、キャラクタの形状が異なりそのエンブレムがキャラクタに貼り付けられたように見せるために、さらに計算を行わなければならないと計算量が多くなるという問題があった。キャラクタに対する固定位置と移動した位置とでポリゴン面が異なる場合があるからである。

【0009】

また、車のキャラクタに対して色や模様を付するような場合、その作成画面は車のキャラクタが平面に展開された画面において行われるため、その画面内で作成して実際に車のキャラクタに貼り付けると、ユーザーの希望する表示とは若干異なる場合がある。実際に車のキャラクタに貼り付けられた表示を予想して作成画面で模様等を作成する必要があるため、何度もやり直したりするなど複雑な操作が必要でかつ熟練した技能を要するという問題があった。

【0010】

そこで、本発明は上記問題点に鑑みてなされたもので、その目的は、ユーザーが作成した表示を、複雑な操作をせずに、キャラクタが複雑な形でも自由にそのキャラクタに貼り付けることのできるプログラム、当該プログラムが格納されたゲーム機、及び当該プログラムが格納された記録媒体を提供することにある。

【0011】

【課題を解決するための手段】

上記目的を達成するために、本発明のプログラムは、画面内で所定の操作を行

うための操作手段と、所定の演算を行う演算処理手段と、操作手段と演算処理手段とに接続され演算処理手段を制御する制御手段とを有するゲーム機において、ゲームのキャラクタに所定の画像を投影するプログラムであって、操作手段の操作により2次元座標からなる画像を制御手段により生成する画像作成ステップと、操作手段からの入力信号に基づいて、画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、画像とを、三次元仮想空間内においてキャラクタ周囲の任意の位置に配置し、投影光源からキャラクタに画像を投影した投影像をキャラクタに貼付するステップとを実現させている。

【0012】

さらに、本発明のかかるキャラクタは、複数のパーツの組み合わせによって構成されており、制御手段に、操作手段の操作に応じてパーツのうち少なくとも一つが画像の投影対象として指定させ、指定されたパーツに対して投影画像を貼り付けさせる、ことを実現させる。

【0013】

また、上記目的を達成するために、本発明は、画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、操作手段と演算処理手段とに接続され演算処理手段を制御する制御手段とを有し、ゲームのキャラクタに所定の画像を投影するプログラムが組み込まれたゲーム機であって、操作手段の操作により2次元座標からなる画像を前記制御手段により生成する画像作成ステップと、操作手段からの入力信号に基づいて、画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、画像とを、三次元仮想空間内においてキャラクタ周囲の任意の位置に配置し、投影光源からキャラクタに画像を投影した投影像を前記キャラクタに貼付するステップと、を実現させるためのプログラムが組み込まれたゲーム機にある。

【0014】

さらに、本発明におけるキャラクタは、複数のパーツの組み合わせによって構成されており、制御手段に、操作手段の操作に応じてパーツのうち少なくとも一つが前記画像の投影対象として指定させ、指定されたパーツに対して投影画像を貼り付けさせる、ことを実現させるプログラムがゲーム機に組み込まれている。

【0015】

また、上記目的を達成するために、本発明は、画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、操作手段と演算処理手段とに接続され演算処理手段を制御する制御手段とを有するゲーム機に対してゲームのキャラクタに所定の画像を投影するプログラムが格納された記録媒体であって、操作手段の操作により2次元座標からなる画像を制御手段により生成する画像作成ステップと、操作手段からの入力信号に基づいて、画像作成ステップで作成した画像をキャラクタに投影するための仮想光源と、画像とを、三次元仮想空間内においてキャラクタ周囲の任意の位置に配置し、投影光源からキャラクタに画像を投影した投影像をキャラクタに貼付するステップと、を実現させるためのプログラムが格納された記録媒体にある。

【0016】

さらに、本発明は、かかる投影像をキャラクタに貼付するステップは、キャラクタを構成する各ポリゴンの α 値を所定の値として演算処理手段で演算されるよう制御手段を制御する、ことを実現させるためのプログラムが格納された記録媒体としている。

【0017】

さらに、本発明は、かかるキャラクタは、複数のパーツの組み合わせによって構成されており、前記制御手段に、前記操作手段の操作に応じて前記パーツのうち少なくとも一つが前記画像の投影対象として指定させ、前記指定されたパーツに対して前記投影画像を貼り付けさせることを実現させるためプログラムが格納された記録媒体にある。

【0018】

上記目的を達成するために、本発明は、三次元仮想空間内において、三次元座標からなるオブジェクトに所定の投影画像を投影したオブジェクト画像をディスプレイ装置に表示するための画像データを生成する画像データ生成手段としてコンピュータを動作させる画像表示制御プログラムであって、画像表示制御プログラムは、制御手段に、オブジェクトに投影される所定の投影画像データを生成させ、仮想空間内にオブジェクトおよび投影画像を配置させ、操作者の操作に基

づいてオブジェクトに対する投影画像の相対位置と、投影画像をオブジェクトに投影する仮想光源の位置を決定させ、仮想光源とオブジェクト上の投影位置を含む投影面の距離を演算し、仮想光源から所定距離以上はなれた投影面を投影対象から除いて、オブジェクトにおいて前記投影画像を投影すべき投影面を演算させ、仮想光源と視点として投影画像を投影面に投影して、投影画像をオブジェクトの前記投影面に貼り付けさせ、投影画像が貼り付けられたオブジェクトについてのオブジェクト画像データを生成させる機能を実現させることを特徴とする画像表示制御プログラムにある。

【0019】

さらに、本発明は、かかる制御手段に、さらに投影画像の画像データをユーザーが操作手段から入力する操作信号に基づいて生成させる機能を実現させることを特徴とする画像表示制御プログラムにある。

【0020】

さらに、本発明は、かかる制御手段に、投影画像をオブジェクトと同一ないし近似する形状を備える透明オブジェクトに対して投影して、投影画像を透明オブジェクトに貼り付けさせ、表示装置に投影画像を貼り付けた透明オブジェクトをオブジェクトに重ねて表示させる機能を実現させることを特徴とする画像表示制御プログラムにある。

【0021】

さらに、本発明は、前記仮想光源と前記投影オブジェクトと前記オブジェクトの位置関係によって、オブジェクトに貼り付けられた投影画像の変形が著しいと判断される場合、投影オブジェクトの透明度を変化させる機能を実現させることを特徴とする画像表示制御プログラムにある。

【0022】

さらに、本発明は、かかるオブジェクトが複数のパーツの組み合わせによって構成されており、制御部に、操作者の操作に応じてパーツのうち少なくとも一つが投影画像の投影対象として指定させ、指定されたパーツに対して投影画像を貼り付けさせる機能を実現させることを特徴とする画像表示制御プログラムにある。

。

【0023】

【発明の実施の形態】

次に本発明の実施の形態について図面を参照して詳細に説明する。

【0024】

図1は、本発明が適用されるゲーム機50の内部構成を示す図である。

【0025】

ゲーム機50は、CPU1と、操作部2と、主記憶装置3と、記録媒体4と、ジオメトリックプロセッサ5と、ローカルバッファ6と、ディスプレイプロセッサ7と、テクスチャバッファ8と、ディスプレイ9とから構成される。

【0026】

CPU1は、操作部2と主記憶装置3と記録媒体4、およびテクスチャバッファ8に接続される。操作部2によるユーザーの各種操作に関する制御信号が入力されるとともに、主記憶装置3や記録媒体4に記憶された各種プログラムを実行するためのものである。その詳細は後述する。

【0027】

操作部2は、CPU1に接続され、ゲーム機50とユーザーとのインターフェースとしての役割を有し、ユーザーが種々の操作を行うことで、後述するエンブレムの作成や、キャラクターの貼り付け、さらにゲーム実行中におけるキャラクターの操作などを行うことができる、そして操作部2は、ユーザーの所定の操作に対応した制御信号をCPU1に出力する。

【0028】

主記憶装置3は、CPU1に接続されるとともにテクスチャバッファ8に接続されている。主記憶装置3には、エンブレム表示のためのプログラムなど各種プログラムが格納されている。また、CPU1で実行されたプログラムに対する各種データも記憶されることになる。

【0029】

記録媒体4には、ゲームの種類ごとに異なるプログラムが記録されており、例えばCD-ROMなどの媒体から構成されている。記録媒体4は、ゲーム機50と着脱可能になっており、記録媒体4をゲーム機50に装着させると、記録媒体

4 に格納されたプログラムが C P U 1 の制御により、C P U 1 を経由して主記憶装置 3 に出力させるようになっている。

【 0 0 3 0 】

ジオメトリックプロセッサ 5 は、C P U 1、ローカルバッファ 6、及びディスプレイプロセッサ 7 に接続されている。ジオメトリックプロセッサ 5 は、C P U 1 から出力される各種データが入力され、このデータをもとに画面表示等を行うための演算を行う。演算した結果、演算データは、ディスプレイプロセッサ 7 に出力される。

【 0 0 3 1 】

ローカルバッファ 6 は、ジオメトリックプロセッサ 5 と接続され、ジオメトリックプロセッサ 5 から出力される演算データを必要に応じて記憶する。

【 0 0 3 2 】

ディスプレイプロセッサ 7 は、テクスチャバッファ 8 と接続されとともに、ディスプレイ 9 に接続されている。ジオメトリックプロセッサ 5 で演算した演算データをもとに実際にディスプレイ 9 に表示させるために演算を行い、画面データとしてディスプレイ 9 に出力する。また、ディスプレイプロセッサ 7 は、ジオメトリックプロセッサ 5 からの演算データをもとに適宜、テクスチャバッファ 8 に出力する。

【 0 0 3 3 】

テクスチャバッファ 8 は、ディスプレイプロセッサ 7 から出力されるテクスチャを一時記憶するとともに、ディスプレイ 9 にも接続され、表示画面（フレーム画面）に対する奥行きを示す Z 値も一時記憶する。この Z 値の記憶については後述する。また、テクスチャバッファ 8 は、C P U 1 及び主記憶装置 3 にも接続され、C P U 1 からの制御信号が入力されて主記憶装置 3 から転送されるデータを一時記憶する。

【 0 0 3 4 】

ディスプレイ 9 は、ディスプレイプロセッサ 7 から出力される画面データをもとに、実際にゲーム画像を表示するためのものである。また、上述の Z 値をテクスチャバッファ 8 に出力する。

【0035】

次に図2を用いてエンブレムを貼り付ける対象となるキャラクタ（オブジェクト）を表示させる描画の処理について説明する。この描画の処理のためのプログラムは記録媒体4に記録され、記録媒体4がゲーム機50に装着された場合にCPU1の制御により読み出され、主記憶装置3に転送されて、主記憶装置3に記憶されていることが前提となる。もちろん、当該プログラムが予めゲーム機50に組み込まれて主記憶装置3に予め記憶されている場合でも適用できる。

【0036】

まず、オブジェクトの描画についての処理が開始される（ステップS10）と、CPU1は、カメラの行列を生成する（ステップS11）。ここでは、グローバル座標系（x、y、z）の所定の位置にオブジェクト（キャラクタ）が存在している場合に、その座標系の任意の位置に実際にディスプレイ9の画面に表示される視点の位置が存在する。この視点の位置に“カメラ”が位置するものとして、そのカメラの位置等を行列式で表す。そして、この行列式がカメラの行列なのである。カメラの行列式に含まれるものは、カメラのグローバル座標系の位置座標の他、グローバル座標系の原点からの角度の値も含まれる。カメラの行列は、例えば4行3列の行列式や、4行4列の行列式で表されるが、もちろんその他n行m列の行列式で表すことも可能である。生成した行列式は、主記憶装置3に出力して主記憶装置3で記憶される。

【0037】

次いで、CPU1は、描画開始のリクエストをジオメトリックプロセッサ5に出力する（ステップS12）。これによりジオメトリックプロセッサ5は、オブジェクト描画のための処理を開始することになる。本実施例では、ディスプレイ9に表示されるゲーム画像の1フレーム単位の時間（1/60秒：1秒で60フレーム表示される場合）ごとにこのリクエストを出力する。オブジェクトを含む画像の描画は1フレーム単位で行われるからである。

【0038】

次いで、CPU1は、ステップS11で生成したカメラの行列式を主記憶装置3から読み出す（ステップS13）。

【0 0 3 9】

次いで、CPU 1 は、モデルの位置の移動のための処理を行う（ステップ S 1 4）。すなわち、ステップ S 1 3 で読み出したカメラの行列式と、オブジェクト（モデル）の位置とから、カメラの位置からの相対的なオブジェクトの位置を演算する処理を行う。具体的には、ともにグローバル座標系で表されたオブジェクトの位置とカメラの位置とから、カメラの位置を原点とした場合のオブジェクトの位置を演算する処理を行う。これにより、カメラの視点からのオブジェクトの相対位置、すなわち、実際にディスプレイ 9 の画面上に表示されるオブジェクトの位置を求めることができる。

【0 0 4 0】

次いで、CPU 1 は、モデルデータの転送を行う（ステップ S 1 5）。具体的には、CPU 1 は、ステップ S 1 4 で演算したオブジェクトの相対位置を示すモデルデータを主記憶装置 3 に出力する。主記憶装置 3 は、このモデルデータを一時記憶する。

【0 0 4 1】

次いで、CPU 1 は、1 フレームに含まれるすべてのオブジェクトに対してオブジェクトの相対位置を演算したか否か判断する（ステップ S 1 6）。まだ、フレーム内に演算していないオブジェクトがあるなら（ステップ S 1 6 で “N o” のとき）、処理は再びステップ S 1 3 に戻り主記憶装置 3 からカメラの行列式を呼び出して上述したカメラからの相対位置を各オブジェクトに対して演算処理を行う。

【0 0 4 2】

1 フレームに含まれるすべてのオブジェクトに対する演算が行われた場合（ステップ S 1 6 で “Y e s” の場合）、CPU 1 は、描画終了のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 1 7）。そして、CPU 1 は、このリクエストの出力をトリガとして主記憶装置 3 に記憶した各オブジェクトのモデルデータを読み出して、ジオメトリックプロセッサ 5 に出力する。これにより、ジオメトリックプロセッサ 5 は、モデルデータをもとに実際にディスプレイ画面 9 上にオブジェクトを表示するための演算処理が行われる。

【0043】

すなわち、ジオメトリックプロセッサ5から出力される演算データをもとに、ディスプレイプロセッサ7は、画面表示のための処理を行い、その出力がディスプレイ9に出力される。そして、ディスプレイ9では画面上でエンブレムを貼り付ける対象のオブジェクトが表示されることになる（ステップS18）。そして、オブジェクト描画のための処理が終了する（ステップS19）。

【0044】

なお、オブジェクトの描画処理（ステップS18）では、ジオメトリックプロセッサ5において各オブジェクトを表現するポリゴンの α 値をすべて“0”（または“1”）にして色表現のない透明なポリゴン面から構成されるオブジェクトを表現している。計算量を減らすためである。

【0045】

この描画の処理により、実際にディスプレイ9に表示されるオブジェクトの例を図3に示す。ディスプレイ画面15のほぼ中央には、ユーザーがゲーム内に使用するキャラクタ16が表示される。このキャラクタ16がエンブレムを貼付するキャラクタである。図面ではわかりにくいだが、実際には透明のキャラクタが表現されている。

【0046】

画面15の下部には、操作方法17が表示される。操作方法17は、ディスプレイ9の画面上における操作と操作部2との対応関係を示すもので、操作部2の操作によりキャラクタ16の位置を変えたりすることができるようになっている。ちなみに、この画面を表示させるには、例えばメインメニューで個人データ等の選択画面を選択することで表示されるようになっている。

【0047】

次に、図3のように表示されたキャラクタに対して所望のエンブレムを投影して貼付する処理について図4を参照して説明するが、この処理が開始される前の段階でエンブレムの作成を行うことが前提となっている

エンブレムの作成画面の例を図5に示す。ここでエンブレムとは、図3に表示されたキャラクタの所望の位置に貼付するイメージである。ユーザーはこの作成

画面で、エンブレムを好みの画像にしたり、編集したりすることができる。この作成画面も、ゲームのメインメニューで個人データ等の所定の選択を行うことで表示されるようになっている。

【0048】

この作成画面（EMBLEM EDIT）は、エンブレム作成ボタン群 2 1 と、エンブレム拡大画面 2 3 と、表示画面 2 4 と、色選択ボタン群 2 5 とから主に構成される。

【0049】

エンブレム作成ボタン群 2 1 は、エンブレム 2 8 を画面内で作成するための複数のボタン 2 1 0 ～ 2 2 1 から構成され、アイコン 2 7 を所定の操作部 2 の操作により画面 2 0 上で選択できるようになっている。

【0050】

ボタン 2 1 0 は、自由に線を拡大画面 2 3 上で描くためのボタンである。アイコン 2 7 の画面内の操作でボタン 2 1 をクリックすると拡大画面 2 3 内で自由に線を描くことができるようになっている。

【0051】

ボタン 2 1 1 は、いわゆる“消しゴム”のためのボタンで、アイコン 2 7 の操作により拡大画面 2 3 で描いた線を消すことができる。

【0052】

ボタン 2 1 2 は、ペイントのためのボタンで、このボタンをクリックして拡大画面 2 3 内でアイコン 2 7 を所定の位置に移動させ、所定の操作を行うことでアイコン 2 7 が位置する所定範囲内で色が付されるようになされている。

【0053】

ボタン 2 1 3 は、四角形を描くためのボタンであり、画面内での操作により拡大画面 2 3 内で四角形を描くことができる。

【0054】

ボタン 2 1 4 は、描いたエンブレム 2 8 をプレビューするためのボタンであり、ボタン 2 1 5 は、作成画面を終了させるためのボタンである。

【0055】

ボタン 2 1 6 は、ペンの太さを設定するためのボタンで、拡大画面 2 3 で描くエンブレムの線の太さを設定することができる。

【 0 0 5 6 】

ボタン 2 1 7 は、拡大画面 2 3 内で描いたエンブレム 2 8 をクリアするためのボタンで、ボタン 2 1 8 は直線を描くためのボタンである。

【 0 0 5 7 】

ボタン 2 1 9 は、円を描くためのボタンであり、ボタン 2 2 0 は、エンブレム 2 8 をさらに拡大させるためのボタンである。そして、ボタン 2 2 1 は、いわゆる “U n d o” 機能のボタンであり、直前の操作を取り消して元に戻すためのボタンである。

【 0 0 5 8 】

拡大画面 2 3 には、エンブレム 2 8 を拡大表示させ、細かな調整を行うことができるようになっている。

【 0 0 5 9 】

表示画面 2 4 は、後述する図 1 0 に示す投影を行う画面で表示されるエンブレムの大きさとほぼ等しい大きさでエンブレム 2 8 が表示される画面である。ユーザーはこの画面を確認しながら拡大画面 2 3 においてエンブレム 2 8 を作成することができる。

【 0 0 6 0 】

色選択ボタン群 2 5 は、複数の色がボタンとして表示されており、アイコン 2 7 を画面 2 0 で移動させ所望の色を示すボタンを選択することで、拡大画面 2 3 に表示されたエンブレム 2 8 に色を付すことができる。

【 0 0 6 1 】

このエンブレム作成画面 2 0 は、以下のようにして表示されることになる。すなわち、記録媒体 4 に記録されたエンブレム作成画面を表示するためのプログラムが格納され、これを C P U 1 が読み出し、表示のための所定のデータをジオメトリックプロセッサ 5 に出力する。ジオメトリックプロセッサ 5 は、表示のための処理を行い、演算データをディスプレイプロセッサ 7 に出力する。ディスプレイプロセッサ 7 は演算データをもとに表示のためのデータを生成し、ディスプレ

イ 9 に出力され、実際のエンブレム作成画面がディスプレイ 9 に表示されることになる。

【0062】

そして、操作部 2 の所定の操作により、その制御信号が CPU 1 に出力されその制御データをもとに CPU 1 は、再びその操作に対応した表示を上述の動作により行う。終了ボタン 215 の操作によりそれまで作成画面 20 で作成したエンブレム 28、すなわち、ジオメトリックプロセッサ 5 でエンブレム 28 の表示の演算を行うためのデータは、主記憶装置 3 に記憶されることになる。以上がエンブレム作成のための処理である。

【0063】

再び図 4 に戻って、上述により作成したエンブレム 28 をキャラクタ 16 に貼付する投影の処理について説明する。かかる投影処理のためのプログラムは記録媒体 4 に記録され、記録媒体 4 をゲーム機 50 に装着させた場合に CPU 1 の制御により読み出され、主記憶装置 3 に転送され、この主記憶装置 3 で記憶される。もちろん、当該プログラムが記録媒体 4 ではなく、主記憶装置 3 に予め記憶されており、当該プログラムが組み込まれた組み込み型のゲーム機 50 でも適用される。

【0064】

投影の処理が開始される（ステップ S30）と、CPU 1 はライト行列の生成を行う。ライト行列の生成とは、グローバル座標系で表示されたキャラクタ 16 に対して、任意の位置（ライト位置）を示す行列式を生成することである。このライト位置からの視点がエンブレム 28 をキャラクタ 16 に貼り付けを行うための画面の視点となる。ちなみに、ここではライトの位置と上述したカメラの位置とは同じである。位置が異なる場合とは、例えば実際にゲームを実行している最中にキャラクタ 16 自身はカメラからの視点、エンブレム 28 はライトからの視点とする場合が考えられる。このようにすることで、エンブレム 28 のみが例えばキャラクタ 16 の側面に実際の画面からの視点とは異なる視点のエンブレム 28 を表示することができるようになる。

【0065】

図6には、グローバル座標系で所定の位置にキャラクタ16とライトの位置30とが存在する図を示す。この場合ライトの位置30は、キャラクタ16に対して右側後方に位置している。このライトの位置30とカメラの位置とが同じ場合には、ライトの位置30からの視点が実際の画面上に表示されることになる。

【0066】

さらに、図6に示すようにライトの位置30から所定の距離離れた位置にエンブレムテクスチャ面31が存在する。これは上述したエンブレム作成画面20（図5参照）で作成したエンブレム28が表示されるテクスチャ面である。

【0067】

ライトの位置とカメラの位置とはこの場合同じであるので、ライト行列は上述したカメラの行列と同様の行列式で表現される。作成したライト行列は主記憶装置3に出力し、一時記憶される。

【0068】

再び図4に戻って、ライト行列を生成したCPU1は、次いで、投影行列を生成する（ステップS32）。この投影行列は、図6に示すエンブレムテクスチャ面31が表示される位置や、テクスチャ面31の大きさを示す行列式である。ここでエンブレムテクスチャ面31の位置は、グローバル座標系ではライトの位置30から一定の距離に位置しているため、この距離やテクスチャ面31の大きさからこの投影行列を作成することができる。エンブレムテクスチャ面31は、初期設定で例えば 0.5×0.5 で表される。0.5はディスプレイ9に表示される画面の半分であり、 0.5×0.5 は画面の縦、横それぞれ半分の大きさを示す。生成した投影行列は、主記憶装置3に出力し、一時記憶される。

【0069】

次いで、CPU1は、ビュー行列の生成を行う（ステップS33）。このビュー行列の生成は、ステップS31で生成したライト行列とステップS32で生成した投影行列とを乗算することで生成できる。すなわち、各ステップで主記憶装置3に記憶されたライト行列と投影行列とをCPU1が読み出し、乗算することで生成される。

【0070】

次いで、CPU1は、カメラの行列を生成する（ステップS34）。上述したようにカメラの行列は、ディスプレイ9の画面上の視点を示す行列式で、ここではライトの位置30とカメラの位置とは一致しているので、ライト行列と同内容の行列式を生成することになる。生成の具体的処理は、上述した図2のオブジェクトの描画処理におけるカメラの行列の生成（ステップS11）と同様である。

【0071】

次いで、CPU1は、ジオメトリックプロセッサ5に描画開始のリクエストを出力する（ステップS35）。このリクエストを出力するタイミングも上述したオブジェクトの描画処理（図2）と同様に、1秒で60フレーム表示できる本実施例のゲーム機50では1/60秒ごと（1フレーム表示される時間ごと）に出力する。

【0072】

次いで、CPU1は、エンブレムテクスチャをテクスチャバッファ8に出力する（ステップS36）。図5で示すエンブレム28は、主記憶装置3に記憶されているので、CPU1は、主記憶装置3からエンブレム28の表示を行うためのデータを読み出し、テクスチャバッファ8に出力し、テクスチャバッファ8に記憶される。

【0073】

次いで、CPU1は、テクスチャ座標の生成を行う（ステップS37）。すなわち、作成したエンブレム28がキャラクタ16の各ポリゴン面に貼付されるように、テクスチャ面31に位置するエンブレムの座標をキャラクタ16の損算する位置に変換する。このテクスチャ座標の生成は、ステップS33で生成したビュー行列とキャラクタ16の3次元座標とを用いて生成する。エンブレム28はテクスチャ面では、2次元座標（いわゆるu-v座標系）であるが、キャラクタ16のポリゴンを形成する各頂点座標は3次元座標であるため、生成するテクスチャ座標は3次元座標として表現される。かかる、テクスチャ座標は、キャラクタ16上の座標に変換された座標であるので、ユーザーの作成したエンブレムがキャラクタ16上に貼付された画像が表示できるようになる。

【0074】

次いで、CPU 1 は、カメラの行列をロードする（ステップ S 3 8）。すなわち、CPU 1 は、ステップ S 3 4 で生成したカメラの行列を主記憶装置 3 から読み出す処理を行う。

【0075】

次いで、CPU 1 は、モデルの位置の移動処理、すなわち、カメラの視点からの相対的なオブジェクト（ここではエンブレム 2 8 が貼付されたキャラクタ 1 6）の位置の変換処理を行う（ステップ S 3 9）。オブジェクトの描画の処理と同様に、カメラの視点がディスプレイ 9 の実際の画面からの視点になり、グローバル座標系では任意の位置にあるカメラの位置に対して、このカメラ位置を例えば原点としてオブジェクトの位置の座標を演算する。

【0076】

次いで、CPU 1 は、ステップ S 3 9 で演算した各データを主記憶装置 3 に出し（ステップ S 4 0）、主記憶装置 3 はこれらのデータ（モデルデータ）を記憶する。

【0077】

次いで、1 フレーム内のすべてのオブジェクトに対する相対位置の演算を行うと、描画終了のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 4 1 で “Y e s” のとき）。描画終了のリクエストを CPU 1 がジオメトリックプロセッサ 5 に出力しないとき（ステップ S 4 1 で “N o” のとき）、すなわち、1 フレームに含まれるすべてのオブジェクトに対する演算を終了していないとき、処理は再びステップ S 3 5 に移行し、上述の処理を繰り返すことになる。

【0078】

そして、CPU 1 は、このリクエストの出力をもとにステップ S 4 0 で主記憶装置 3 に記憶したモデルデータを読み出して、ジオメトリックプロセッサ 5 に出力する。これにより、ジオメトリックプロセッサ 5 は、実際にディスプレイ画面 9 上にエンブレムが貼付されたオブジェクトを表示するための処理が行われる。さらに、ジオメトリックプロセッサ 5 から出力される演算データからディスプレイプロセッサ 7 は、画面表示のための処理を行い、その出力がディスプレイ 9 に出力される。そして、ディスプレイ 9 では画面上でエンブレムが貼付されたオブ



ジェクトが表示されることになる（ステップS42）。そして、オブジェクト描画のための処理が終了する（ステップS43）。

【0079】

この投影の処理により実際にディスプレイ9上に表示される、オブジェクトの例を図7、図8に示す。図7は、キャラクタ16にエンブレムを貼り付ける直前の画面であり、処理としてはテクスチャ座標の生成を行っていない（ステップS37の処理が行なわれていない場合）もので、図8はエンブレムをキャラクタ16に貼り付けた場合（ステップS37でテクスチャ座標の生成が行われた場合）の画面を示す。

【0080】

図7は、エンブレム28がまだキャラクタ16に貼り付いていないので、エンブレム28は、図5に示すエンブレム作成画面20の実際の表示画面24に表示されたエンブレム28がそのまま画面15に表示されている。画面15の下部には操作方法17が表示され、この記述に基づいてユーザーは操作部2を操作することで、エンブレム28の位置を移動させたり、その大きさを変化させたりすることができる。

【0081】

操作部2によってエンブレムの大きさを変化させたとき、その変化に対応する制御信号が操作部2からCPU1に出力され、この制御信号に基づいてCPU1は、上述の投影の処理（図4）のステップS32の投影行列の生成で、その大きさに対応した投影行列を生成する。すなわち、大きさが変化すると初期設定の0.5の値が変化してその大きさの値に対応する値を用いて投影行列を生成して上述の投影の処理を行うことになる。

【0082】

また、エンブレムの位置を変化させたいときは、操作部2によって所定の操作により行われる。すなわち、その変化させた位置に対応する制御信号が操作部2からCPU1に出力される。そしてCPU1は、この制御信号に基づいて画面の視点位置、すなわちグローバル座標系のライトの位置や原点からの角度が変化し、その移動した位置に対応するライト行列をステップS31生成して以後上述の

処理（図4参照）を行って、エンブレム28の移動を行うことができる。

【0083】

また、キャラクタ16の表示位置を変えたい場合は、上述したオブジェクトの描画処理（図2）で表示された画面に対して、操作部2での所望の操作により行われる。すなわち、その操作に対応する制御信号が操作部2からCPU1に出力され、この制御信号に基づいてCPU1は、変化させた表示位置に対応するカメラの行列をステップS11で生成することで、以後の処理を行うことで実現できる。

【0084】

そして、図7に表示されたエンブレム28のキャラクタ16に対する位置や大きさをユーザーが画面15内で確認して、これによれば操作部2に所定の操作を行うことで、上述の投影の処理（図4）がすべて行われて図8に示す実際のキャラクタ16にエンブレム28が貼り付いた状態で画面表示される。

【0085】

このキャラクタ16の視点位置の移動も同様にカメラ行列又はライト行列を移動した位置に対応した行列式を生成する（ステップS30、ステップS34）ことで実現できる。

【0086】

ここで、エンブレムの貼り付けについて、1つあるいは複数のポリゴンから構成されるキャラクタ16の各パーツごとに行うことも可能である。

【0087】

また、表示されるキャラクタ16は、透明に表示されるが、キャラクタ16上の貼り付け位置は、最も手前、すなわちカメラからもっとも近いポリゴン面に行うようにしている。

【0088】

以上のようにして、エンブレム28のキャラクタ16への投影を行うことができる。したがって、画面を見ながら貼り付けたい位置に自由にエンブレム29をキャラクタ16貼り付けることができる。また、エンブレム28が貼付されたキャラクタ16を貼付直後に確認できるので、熟練した技能を必要とせず、だれで

も所望のエンブレムを貼付することができる。さらに、操作部 2 の所定の操作により投影が行われるため、複雑な操作を必要としない。

【0089】

次に、実際にゲームを行う場合に、エンブレム 28 が貼付されたキャラクタ 16 を含むゲームの表示の処理について図 9 から図 11 を用いて説明する。かかる表示処理のためのプログラムは記録媒体 4 に記録され、この記録媒体 4 をゲーム機 50 に装着されると CPU 1 の制御により主記憶装置 3 に転送され、主記憶装置 3 にて記憶され、以後の処理が行われる。もちろん、予めこの処理のためのプログラムが主記憶装置 3 に記憶されている、プログラム組み込み型のゲーム機でも適用される。

【0090】

まず、ゲーム表示の処理が開始される（ステップ S 50）と、CPU 1 は、ライト行列を生成する（ステップ S 51）。上述した投影処理（図 4）のライト行列の生成（ステップ S 31）と同様である。すなわち、グローバル座標系での所定の位置（ライトの位置）を行列式で表現したものを生成する。同様に m 行 n 列で表現できる。生成したライト行列は主記憶装置 3 に出力され、一時記憶される。

【0091】

次いで、CPU 1 は、投影行列を生成する処理を行う（ステップ S 52）。上述の投影処理（図 4）の投影行列の生成（ステップ S 32）と同様である。すなわち、図 6 で参照したようにライトの視点位置 30 からエンブレムテクスチャ面 31 までの距離が一定であるため、エンブレムテクスチャ面 31 のグローバル座標系の位置、テクスチャ面 31 の大きさなどを行列式で表現したものを生成する。生成した投影行列は主記憶装置 3 に出力され、一時記憶される。

【0092】

次いで、CPU 1 は、ビュー行列を生成する処理を行う（ステップ S 53）。すなわち、上述のステップ S 51 及び S 52 で生成したライト行列と投影行列とを主記憶装置 3 から読み出してこれらを乗算してビュー行列を生成する。生成したビュー行列は主記憶装置 2 に出力され、一時記憶される。

【0093】

次いで、CPU1は、デプス行列を生成する（ステップS54）。これは、予め主記憶装置3に記憶されたデプス用の行列とステップS51で生成したライト行列とを乗算して生成する。生成したデプス行列は、主記憶装置3に出力され一時記憶される。

【0094】

次いで、CPU1は、現在のカメラの位置から投影行列を生成する（ステップS55）。生成した投影行列は、主記憶装置3に出力され一時記憶される。また、CPU1は、現在のカメラ位置からカメラ行列を生成する。生成したカメラ行列は主記憶装置3に出力され一時記憶される。

【0095】

次いで、CPU1は、描画開始のリクエストをジオメトリックプロセッサ5に出力する（ステップS56）。これにより、ジオメトリックプロセッサ5は、描画のための処理の準備に移行する。ここでは、いわゆるプリレンダリングの処理のための描画開始のリクエストである。

【0096】

次いで、CPU1は、ステップS55で生成したカメラ行列を主記憶装置3から読み出す（ステップS57）。

【0097】

次いで、CPU1は、オブジェクト（モデル）を表示したい位置に移動させる（ステップS58）。具体的には、オブジェクトの描画の処理（図2参照）のステップS14と同様に、ともにグローバル座標系で表されたオブジェクトの位置とカメラの位置とから、カメラの位置を原点とした場合のオブジェクトの位置を演算する処理を行う。これにより、カメラの視点からのオブジェクトの相対位置、すなわち、実際にディスプレイ9の画面上に表示されるオブジェクトの位置を求めることができる。

【0098】

次いで、CPU1は、モデルデータの転送を行う（ステップS59）。具体的には、CPU1は、ステップS58で演算したオブジェクトの相対位置を示すモ

デルデータを主記憶装置 3 に出力され、一時記憶される。

【0099】

次いで、CPU 1 は、描画終了のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 6 0）。そして、CPU 1 は、主記憶装置 3 に記憶されたモデルデータを読み出してジオメトリックプロセッサ 5 に出力してオブジェクトの描画のための処理が行われる（ステップ S 6 0）。

【0100】

次いで、CPU 1 は、図 10 のステップ S 6 2 に移行して、描画結果の Z 値をテクスチャバッファ 8 に出力させるようジオメトリックプロセッサ 5 に指示データを出力する。ここでディスプレイ 9 の画面上にオブジェクトが表示されている場合に、画面上の任意の点（カメラの位置からの相対的なオブジェクトの位置）は 2 次元座標で表されているものの、グローバル座標系では 3 次元座標として表されている。そこで、画面上の任意の画素ごとにグローバル座標系でオブジェクトが表された場合の Z 値、すなわち奥行きを表す値をテクスチャバッファ 8 に記憶されるよう、CPU 1 は指示データを出力する。この指示データを受けたジオメトリックプロセッサ 5 は、オブジェクトの描画のための演算処理を行っているときに、各 Z 値を演算してディスプレイプロセッサ 7 にその Z 値を出力する。Z 値はディスプレイプロセッサ 7 からディスプレイ 9 を経由してテクスチャバッファ 8 に出力される。そして、テクスチャバッファ 8 において Z 値が保存されることになる。

【0101】

次いで、CPU 1 は、レンダリングのため描画開始のリクエストをジオメトリックプロセッサ 5 に出力する（ステップ S 6 3）。

【0102】

次いで、CPU 1 は、主記憶装置 3 に記憶したカメラ行列を読み出し（ステップ S 6 4）、カメラの位置からの相対的なオブジェクトの位置を演算する（ステップ S 6 5）。具体的には、ステップ S 5 8 と同様とともにグローバル座標系で表されたオブジェクトの位置とカメラの位置とから、カメラの位置を原点とした場合のオブジェクトの位置を演算する処理を行う。そして、CPU 1 は、演算し

たモデルデータを主記憶装置 3 に出力し（ステップ S 6 6）、モデルデータは主記憶装置 3 に記憶される。

【0103】

次いで、CPU 1 は、ステップ S 5 4 で生成したデプス行列を主記憶装置 3 から読み出す（ステップ S 6 7）。

【0104】

次いで、CPU 1 は、主記憶装置 3 に予め記憶された勾配テクスチャとステップ S 5 4 で生成したデプス行列とを主記憶装置 3 から読み出して、デプス値を生成する。ここでデプス値とは、投影の処理（図 4 参照）のステップ S 3 9 で生成した α 値が所定の値（たとえば“0”）である透明なモデルに対する奥行きを示す値である。生成したデプス値は主記憶装置 3 に出力され一時記憶される。

【0105】

次いで、CPU 1 は、ステップ S 5 3 で生成したビュー行列とステップ S 6 2 で生成した Z 値とをそれぞれ主記憶装置 3 とテクスチャバッファ 8 から読み出す（ステップ S 6 9）。

【0106】

次いで、CPU 1 は、ステップ S 6 8 で生成したデプス値とステップ S 6 9 で読み出した Z 値とを比較する（ステップ S 7 0）。デプス値は、演算により生成したモデルの奥行きを示す値で、Z 値は実際にディスプレイ表示されたモデルの奥行きを示す値である。これらを比較することで、画面から見てどちらが手前に位置しているかを判断する。例えば、Z 値の方が大きければ白、そうでなければ黒を透明なモデルに描画する。

【0107】

次いで、CPU 1 は、主記憶装置 3 に記憶されたエンブレムのデータがテクスチャバッファ 8 にコピーされるよう主記憶装置 3 とテクスチャバッファ 8 を制御する（ステップ S 7 1）。したがって、ユーザーが作成したエンブレムの表示のためのモデルデータは主記憶装置 3 とテクスチャバッファ 8 に記憶されることになる。

【0108】

次いで、CPU1は、エンブレムテクスチャ座標の生成を行う（ステップS72）。この生成も投影の処理（図4参照）と同様に、作成したエンブレムがオブジェクトの各ポリゴン面に貼付されるように、エンブレムの座標を変換する。このテクスチャ座標の生成は、ステップS53で生成したビュー行列とオブジェクトの3次元座標とを用いて生成する。エンブレムはテクスチャ面31では、2次元座標（いわゆるu-v座標系）であるが、オブジェクトのポリゴンを形成する各頂点座標は3次元座標であるため、生成するテクスチャ座標は3次元座標として表現される。かかる、テクスチャ座標は、オブジェクトの座標に変換された座標であるので、ユーザーの作成したエンブレムがキャラクタ上に貼付された画像が表示できるようになる。

【0109】

次いで、CPU1は、フェードアウト用行列の生成を行う（ステップS73）。フェードアウト用の行列とは、エンブレムがキャラクタ上に貼付された場合に、例えばライトの位置によってはエンブレムが一定の直線や曲線をならず、のこぎりの刃のように鋭角的な形状が繰り返される場合がある。そこで、かかる形状をオブジェクト上で目立たなくする処理をおこなうためにこの行列を生成する。このフェードアウト用の行列生成のためのデータは予め主記憶装置3に記憶されており、これをCPU1が読み出すことでこの行列を生成する。

【0110】

次いで、CPU1は、図11にステップS74に移行して、フェードアウト用行列からテクスチャ座標を生成する。具体的には、上述のステップS73で生成したフェードアウト用行列と各オブジェクトのポリゴン面の法線ベクトルとからフェードアウト用の座標を生成する。

【0111】

次いで、CPU1は、バックフェースのテクスチャを主記憶装置3から読み出す（ステップS75）。バックフェースとは、オブジェクトのうちディスプレイ9の画面上からは見えない部分のオブジェクトである。かかるバックフェースのテクスチャは、予め主記憶装置3に記憶されている。

【0112】

次いで、CPU 1は、色消しの処理を行う（ステップS 76）。上述のステップS 75で読み出したバックフェーステクスチャに対して α 値を所定値にしたり、黒にしたりすることで色消しを行う。

【0113】

次いで、CPU 1は、透明モデルに対するカメラ位置を演算するため、主記憶装置3に記憶されたカメラの行列を読み出す（ステップS 77）。そして、CPU 1は、読み出したカメラの行列をもとにカメラ位置からの相対的なオブジェクトの位置を演算する（ステップS 78）。演算したオブジェクトの位置は一旦主記憶装置3に出力して一時記憶させる（ステップS 79）。

【0114】

次いで、CPU 1は、1フレーム分の処理が終了すると描画終了のリクエストをジオメトリックプロセッサ5に出力する（ステップS 80で“Y e s”のとき）。まだ1フレーム分の処理が終了していないとき（ステップS 80で“N o”のとき）は、処理は再びステップS 63に移行して上述の処理を繰り返す。

【0115】

描画終了のリクエストを出力した場合には、CPU 1は、ステップS 81に処理が移行し、オブジェクトの描画の処理を行う（ステップS 81）。具体的には、CPU 1は、ステップS 59、ステップS 66、及びステップS 79で主記憶装置3に記憶されたモデルデータを読み出しジオメトリックプロセッサ5に出力し、実際にエンブレムが貼付されたオブジェクトを含む画面全体の描画をするための演算を行う。

【0116】

演算したデータは、ディスプレイプロセッサ7からディスプレイ9に出力され、実際にディスプレイ9に表示されることになる。これにより、実際にゲーム実行中、エンブレム28が貼付されたキャラクタ16を含む、1フレームの画像を表示することができる。そして、上述した処理を繰り返すことで連続した画面表示を行うことができる。

【0117】

ディスプレイに表示された例を図12に示す。画面15内にはユーザーが操作

を行うキャラクタ（オブジェクト）16が位置し、上述したエンブレム28が、キャラクタ16の背面に実際に貼り付けられたように位置している。

【0118】

【発明の効果】

以上説明したように本発明によれば、ユーザーが作成した表示を、複雑な操作をせずに、キャラクタが複雑な形でも自由にそのキャラクタに貼り付けることのできるプログラムを提供することができる。

【図面の簡単な説明】

【図1】

図1は、本発明が適用されるゲーム機の内部構成を示す図である。

【図2】

図2は、オブジェクトを描画するための処理を示すフローチャートである。

【図3】

図3は、エンブレムを貼り付ける対象のオブジェクトを画面表示した場合の一例を示す。

【図4】

図4は、投影の処理を示すフローチャートである。

【図5】

図5は、エンブレム作成画面の一例を示す図である。

【図6】

図6は、投影の概念を説明するための図である。

【図7】

図7は、実際に投影を行う直前の画面の一例を示す図である。

【図8】

図8は、実際に投影を行った後の画面の一例を示す図である

【図9】

図9は、ゲーム内での表示全体の処理を示すフローチャートである。

【図10】

図10は、ゲーム内での表示全体の処理を示すフローチャートである。

【図 1 1】

図 1 1 は、ゲーム内での表示全体の処理を示すフローチャートである。

【図 1 2】

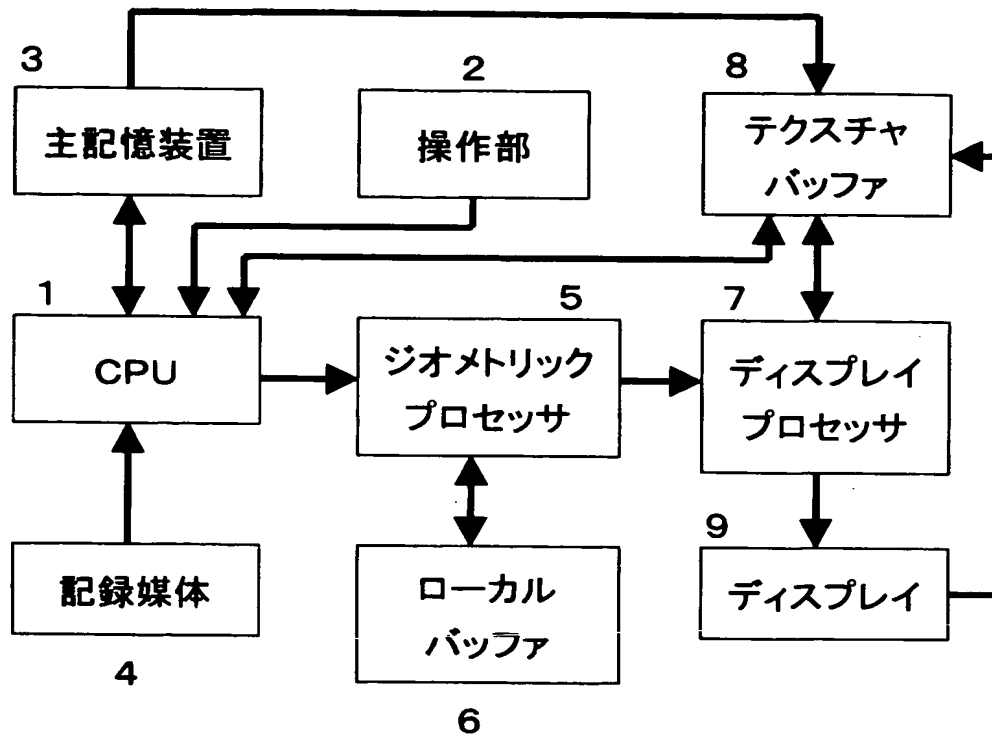
図 1 2 は、ゲームが実行されている場合の画面表示の一例を示す図である。

【符号の説明】

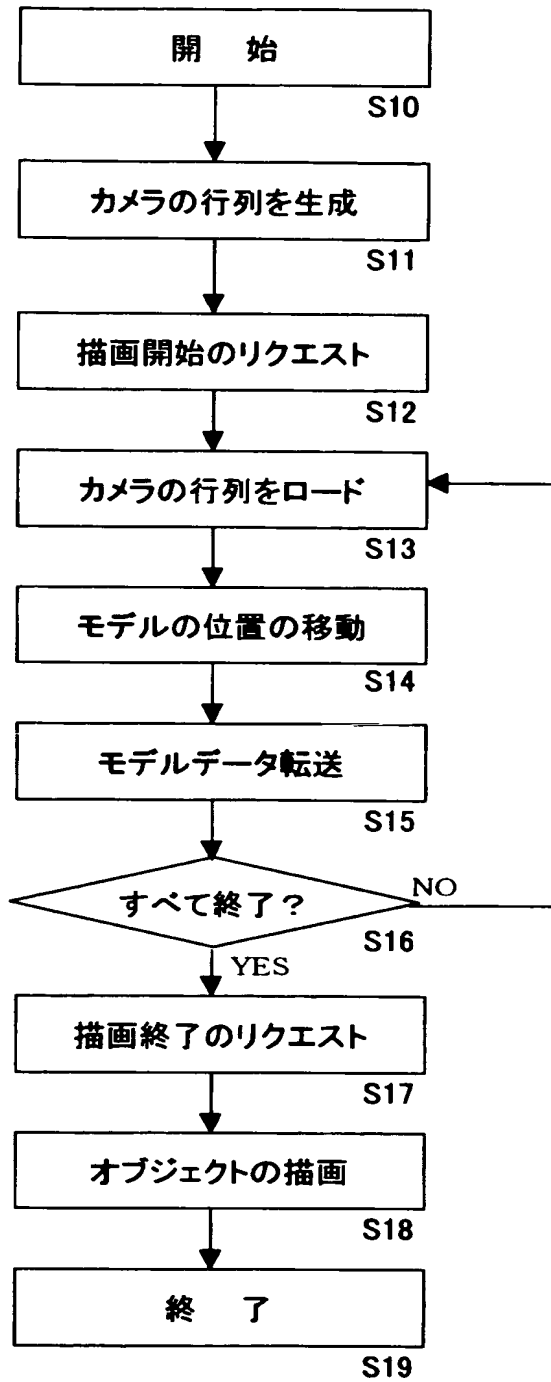
- 1 CPU
- 2 操作部
- 3 主記憶装置
- 4 記録媒体
- 5 ジオメトリックプロセッサ
- 7 ディスプレイプロセッサ
- 8 テクスチャバッファ
- 9 ディスプレイ
- 15 画面
- 16 キャラクタ（オブジェクト）
- 20 エンブレム作成画面
- 21 エンブレム作成ボタン群
- 23 拡大画面
- 24 実際の画面
- 25 色選択ボタン群
- 30 ライトの位置
- 31 エンブレムテクスチャ面

【書類名】 図面

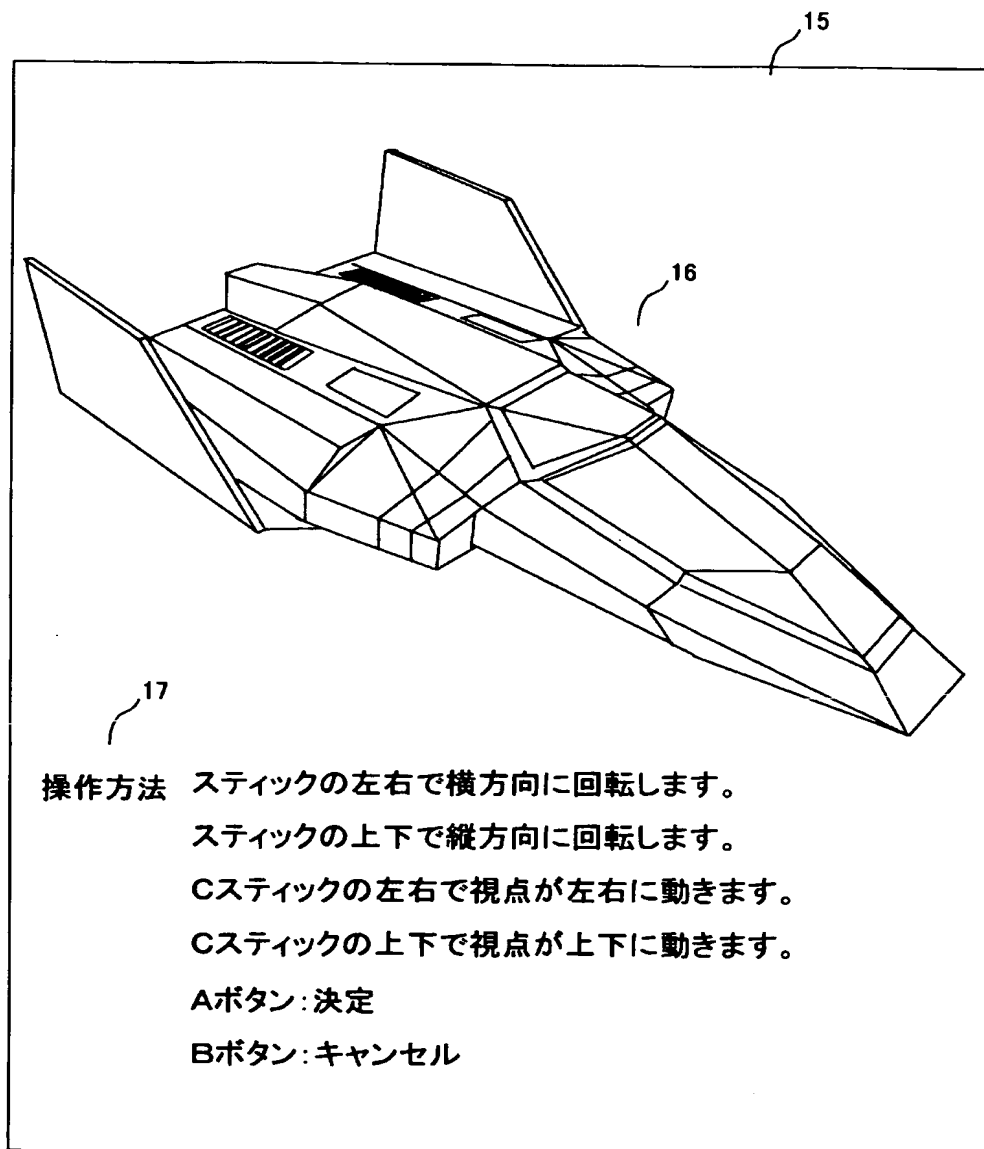
【図 1】

50

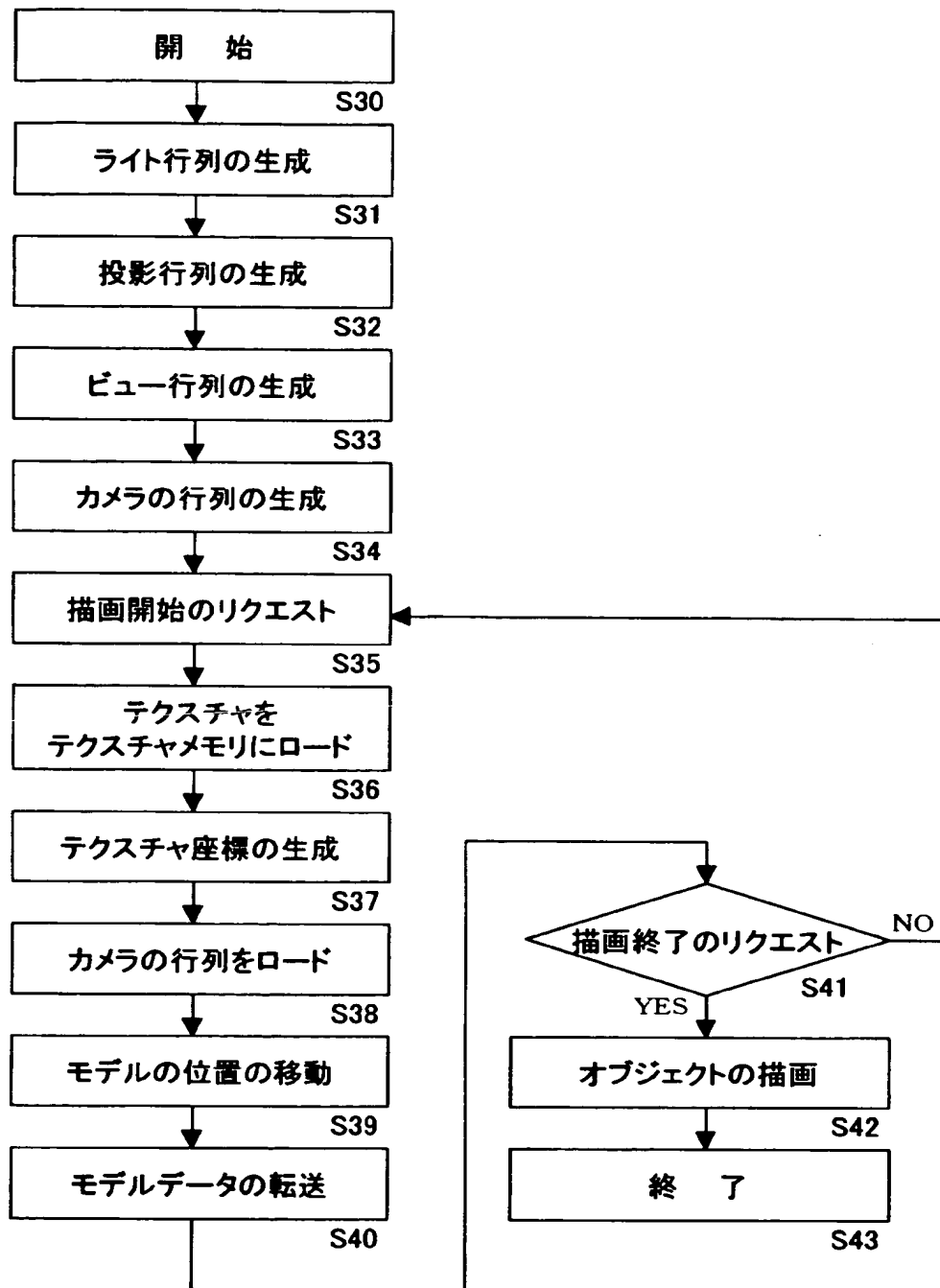
【図 2】



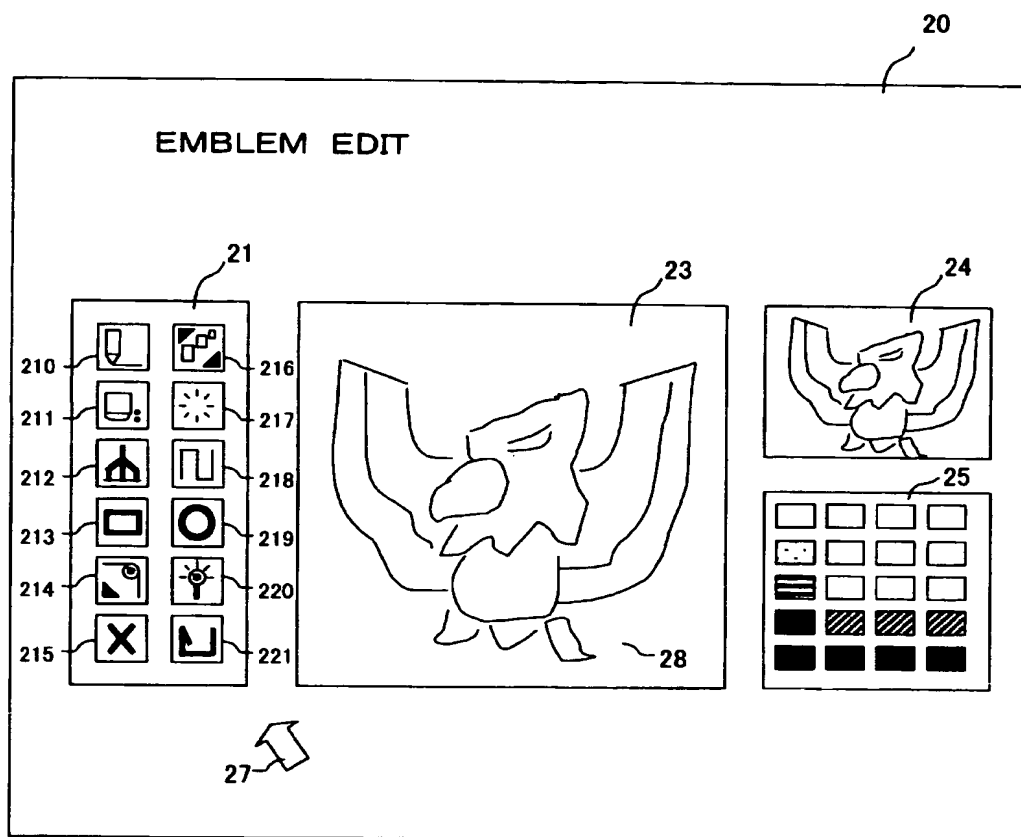
【図 3】



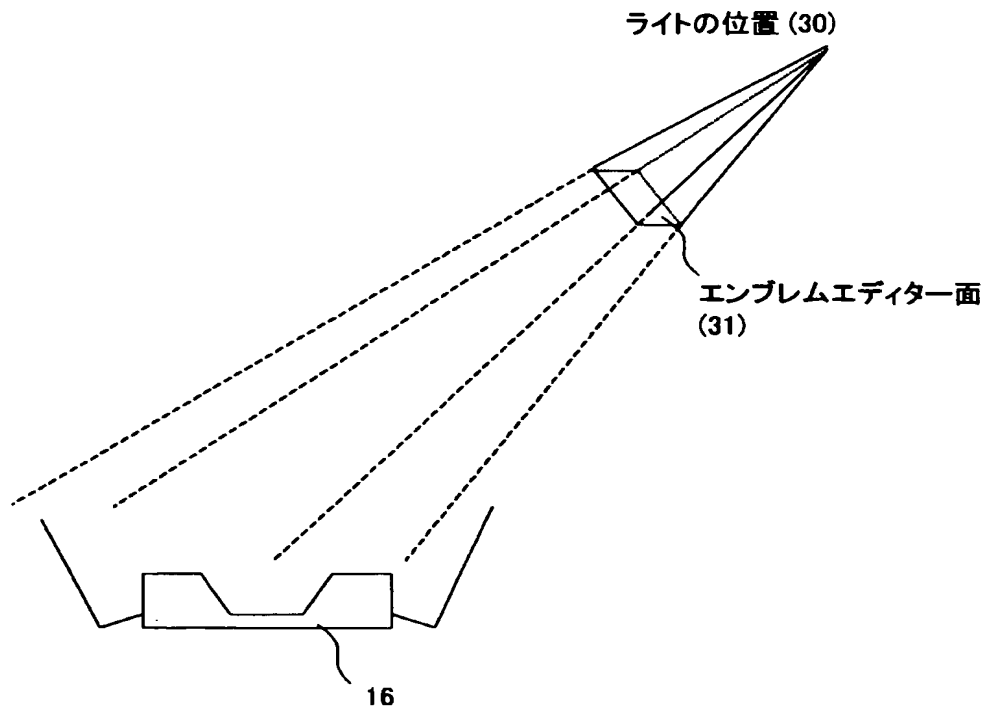
【図 4】



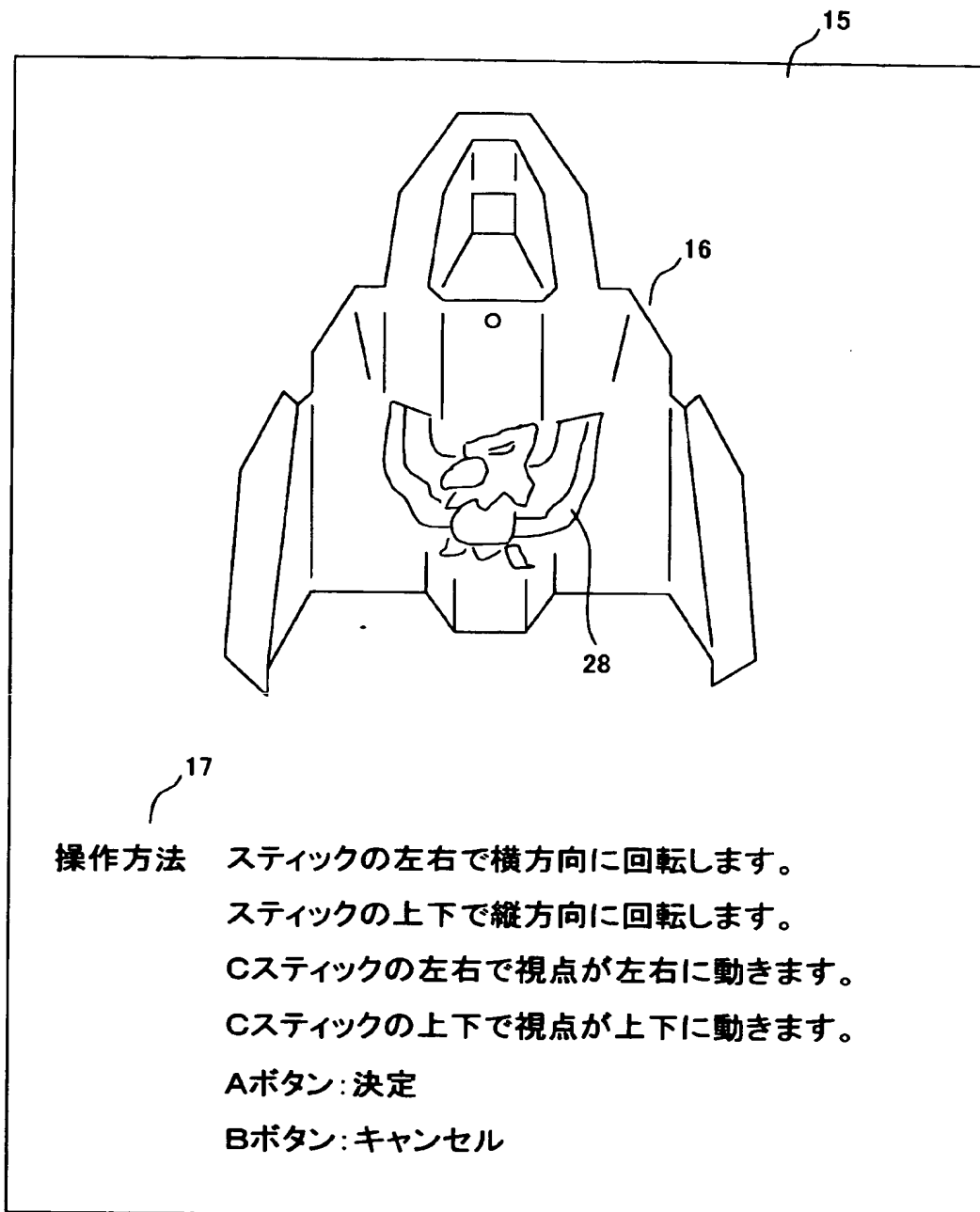
【図 5】



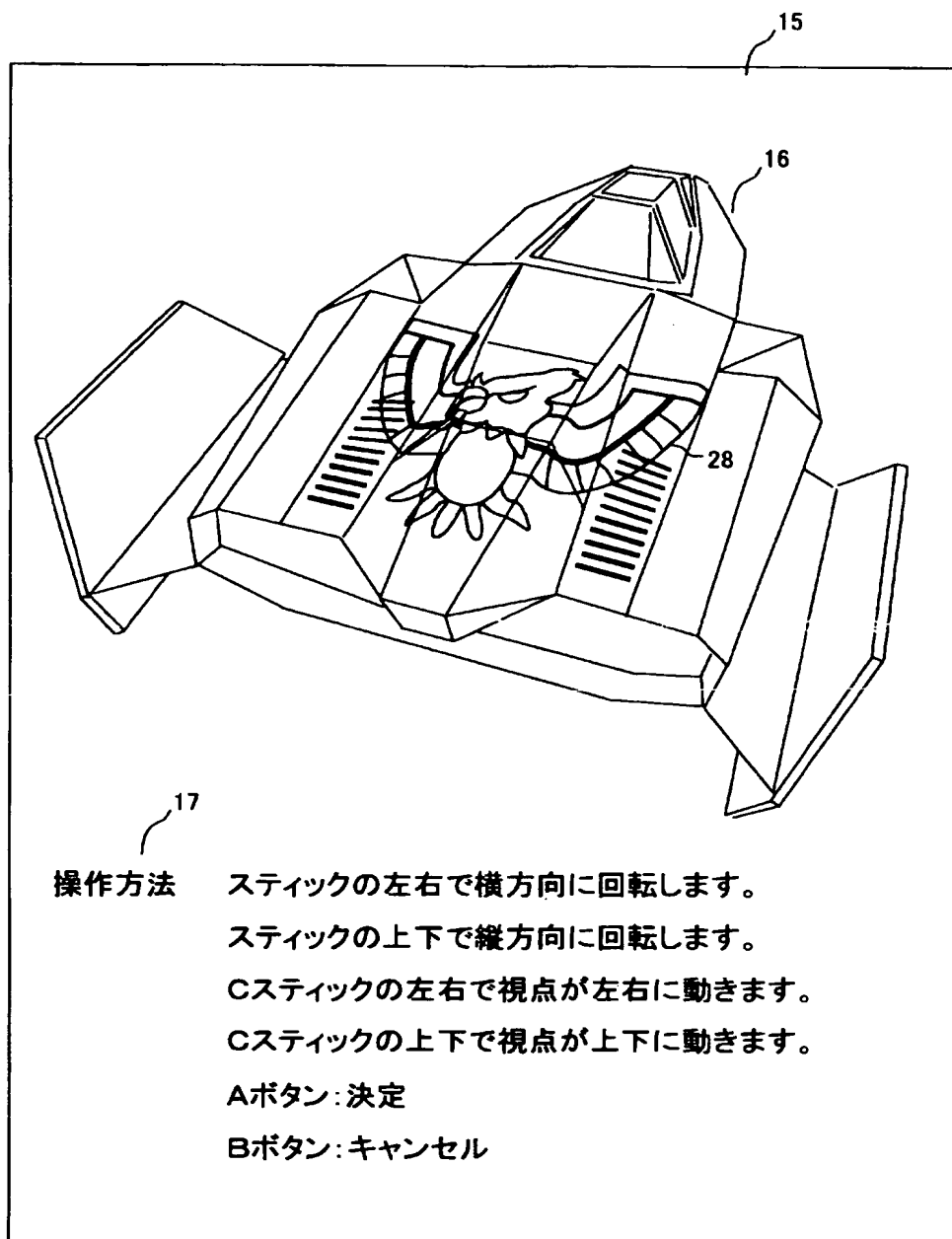
【図 6】



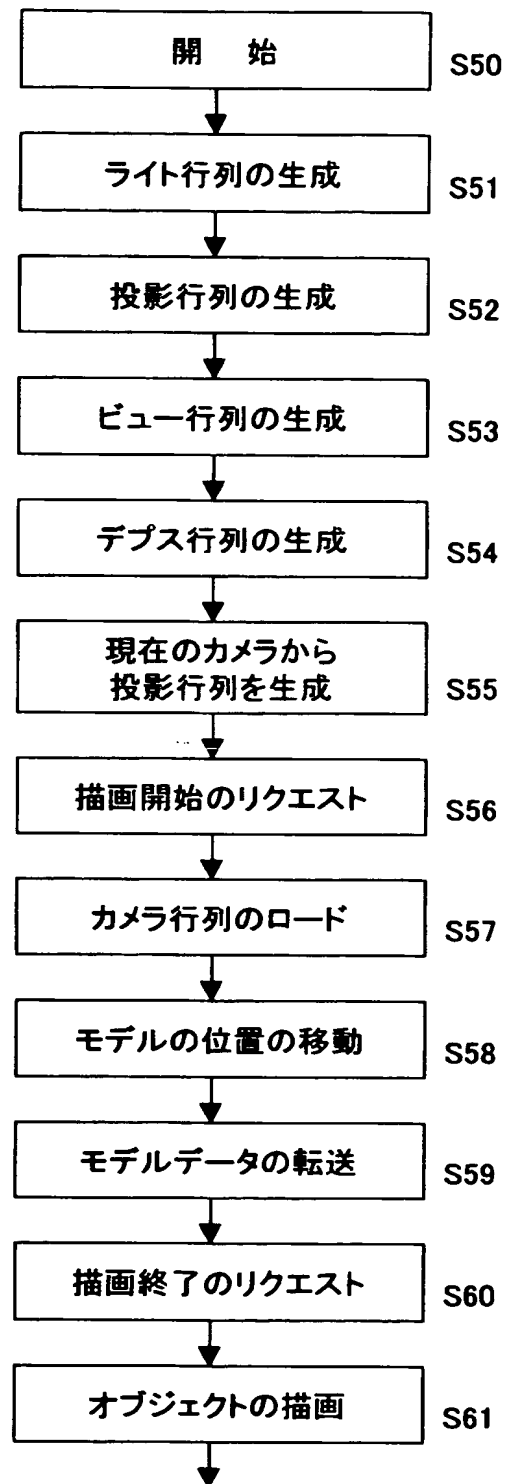
【図 7】



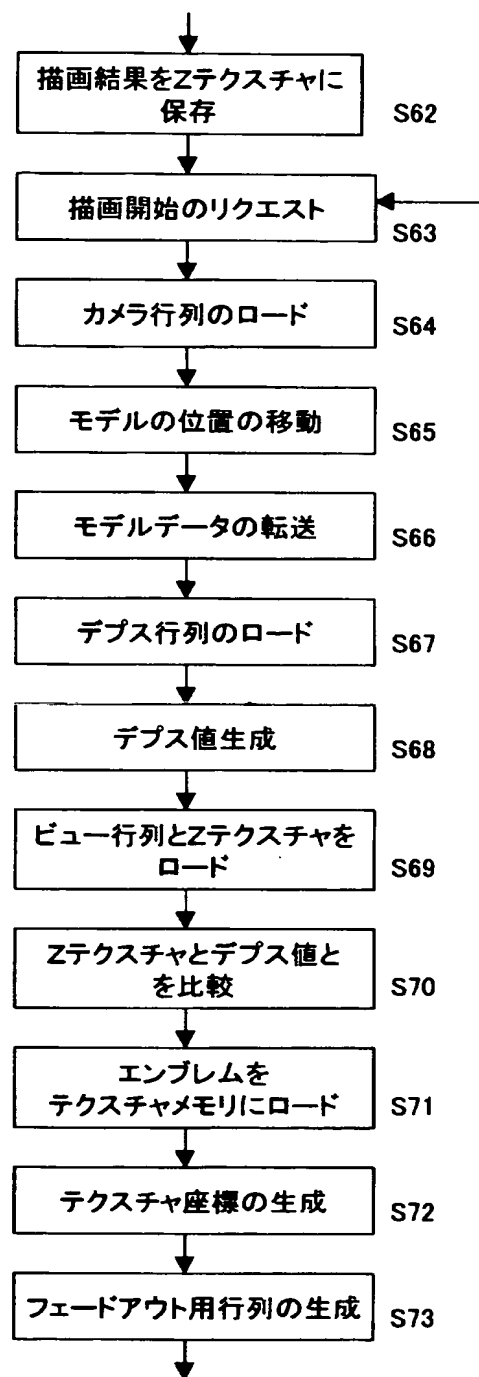
【図 8】



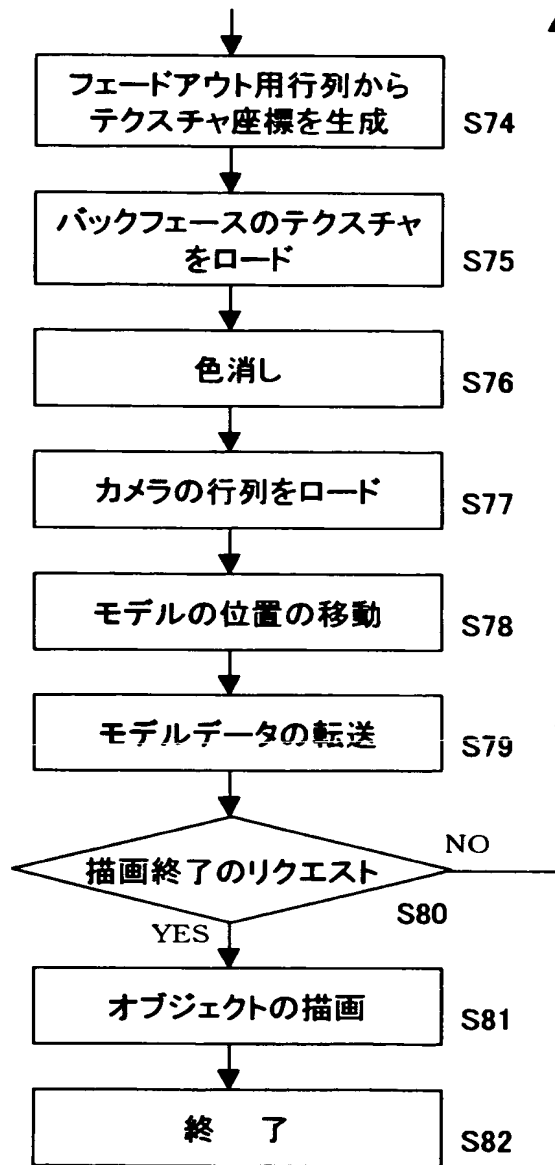
【図 9】



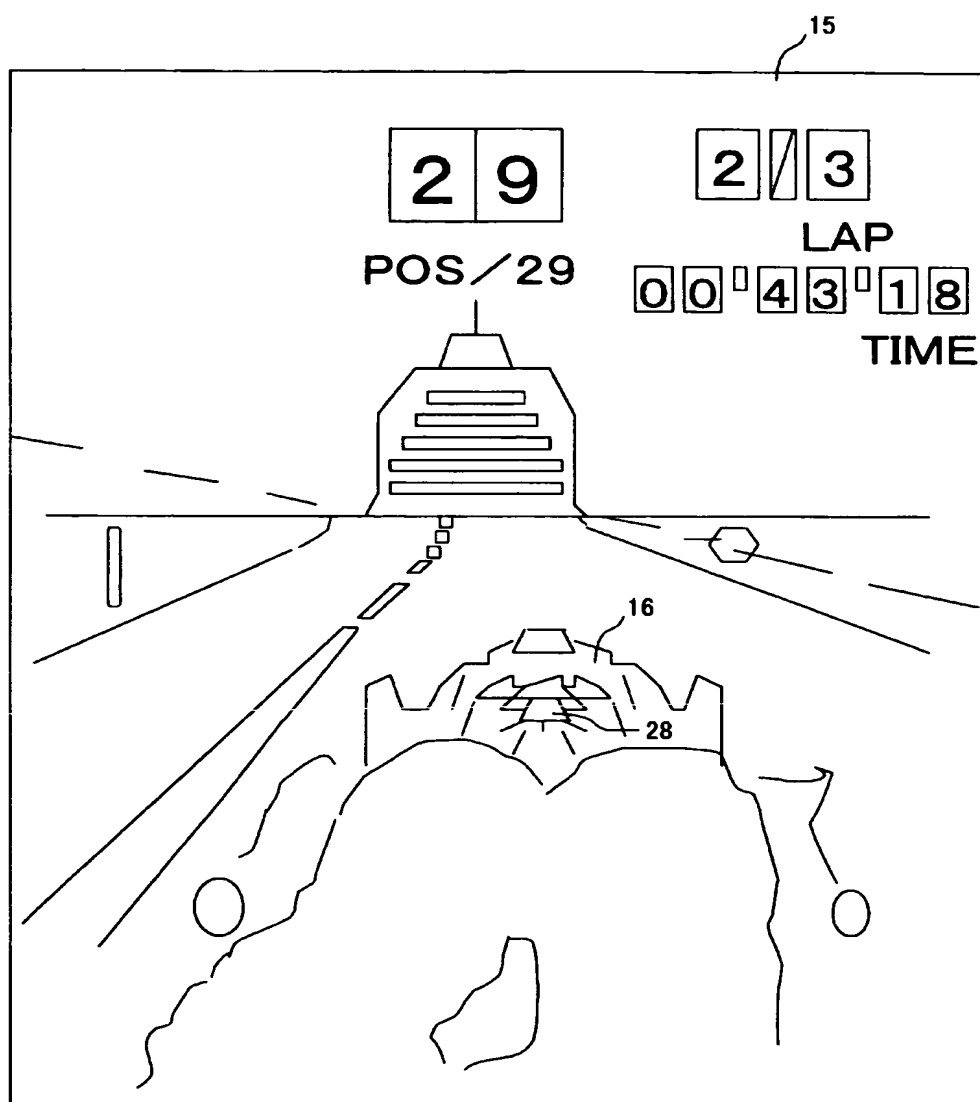
【図 10】



【図 11】



【図 12】



【書類名】**要約書****【要約】**

【課題】 ユーザーが作成した表示を、複雑な操作をせずに、キャラクターが複雑な形でも自由にそのキャラクターに貼り付けることのできるプログラムを提供する。

【解決手段】 画面内で所定の操作を行うための操作手段と、所定の演算を行う演算処理手段と、前記操作手段と前記演算処理手段とに接続され前記演算処理手段を制御する制御手段とを有するゲーム機において、ゲームのキャラクターに所定の画像を投影するプログラムであって、前記操作手段の操作により2次元座標からなる画像を前記制御手段により生成する画像作成ステップと、前記操作手段からの入力信号に基づいて、前記画像作成ステップで作成した画像をキャラクターに投影するための仮想光源と、前記画像とを、三次元仮想空間内において前記キャラクター周囲の任意の位置に配置し、前記投影光源から前記キャラクターに前記画像を投影した投影像を前記キャラクターに貼付するステップと、を実現させるためのプログラム。

【選択図】 図4



特願 2 0 0 3 - 0 1 6 6 3 8

出 願 人 履 歴 情 報

識別番号 [0 0 0 1 3 2 4 7 1]

1. 変更年月日	2 0 0 0 年 1 1 月 1 日
[変更理由]	名称変更
住 所	東京都大田区羽田 1 丁目 2 番 1 2 号
氏 名	株式会社セガ

特願 2 0 0 3 - 0 1 6 6 3 8

出 願 人 履 歴 情 報

識別番号

[0 0 0 2 3 3 7 7 8]

1. 変更年月日

2 0 0 0 年 1 1 月 2 7 日

[変更理由]

住所変更

住 所

京都府京都市南区上鳥羽鉾立町 1 1 番地 1

氏 名

任天堂株式会社